

EX 1.A: Program to find the sum of individual digits of a positive integer

Aim:

To write a Java program to find the sum of individual digits of a positive integer.

Algorithm:

1. Start
2. Get the number
3. Declare a variable to store the sum and set it to
4. Repeat the next two step still the number fest
5. Get the rightmost digit of the number with help of the remainder '%' operator by dividing it by10 and adding it to the sum.
6. Divide the number by 10 with the help of '/' operator to remove the rightmost digit.
7. Print or return the sum
8. Stop

Program:

```
import java.io.*;
import java.util.Scanner;
public class ex1
{
    public static void main(String Args[ ]) {
        Int r,n,sum=0;
        Scanner ss=new Scanner(System.in);
        System.out.print("Enter a number:");
        n=ss.nextInt();
        while(n>0) {
            r=n%10;
            sum=sum+r;
            n=n/10;
        }
        System.out.println("The sum of digits of given number is "+sum);
    }
}
```

DATE:

NAME:SHWETA DARSHINI P
REGNO:412521104139.

Output:

```
Enter a number:13965  
The sum of digits of given number is 24
```

Result:

Thus, the java program written to find the sum of individual digits of a positive integer is executed successfully and output is verified.

EX 1.B:Program to generate the first n terms of the sequence.**Aim:**

To write a java program to generate the first terms of the sequence.

Algorithm:

1. Start.
2. Get the Number Of Odd Numbers To Be Printed(n).
3. Declare an Integer Variable(i).
4. Set the Value Of i as 1.
5. Repeat the next three steps till is less than or equal to n.
6. If the Numbers Divisible By 2, skip the remaining steps in the current iteration.
7. If not, print the value.
8. Increment the Value Of By1.
9. Stop.

Program:

```
import java.util.*;
Class oddNums {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
System.out.print("Enter a number: ");
int n=sc.nextInt();
int i,c=0;
System.out.println("Odd numbers are:");
for(i=1;;i++) {
if(i%2 != 0){
c=c+1;
System.out.println(i);}
if(c==n)
{
break;}
}
}
```

DATE:

NAME:SHWETA DARSHINI P
REGNO:412521104139.

}

Output:

```
Enter a number:5
Odd numbers are:
1
3
5
7
9
```

Result:

Thus, the java program to generate the first n terms of the sequence is executed successfully and output is verified.

EX 1.C: Java program to generate all the prime numbers between 1 and n, where n is a value supplied by the user.

Aim:

To write a java program to generate all the prime numbers between 1 and n, where n is a value supplied by the user.

Algorithm:

1. start
2. set ct=0,n=0,i=1,j=1
3. repeat step 4 to 12 until n<10
4. j=1
5. ct=0
6. repeat step 7 to 9 until j<=i
7. if i%j==0 then
- 8: ct=ct+1
- 9: j=j+1
- 10: if ct==2 then print i
- 11: n =n+1
- 12: i = i+1
- 13: end.

Program:

```
import java.util.*;
Class primeNos {
public static void main(String Args[]) {
Scanner sc = new Scanner(System.in);
Int i,j,n,c;
System.out.println("Enter the number till which you want prime numbers: ");
n=sc.nextInt();
System.out.println("Prime numbers are:");
for(i=2;i<=n;i++) {
c=0;
for(j=1;j<=i;j++) {
if(i%j==0) {
```

DATE:

NAME:SHWETA DARSHINI P
REGNO:412521104139.

```
c++;  
}  
if(c==2) {  
System.out.print(i+" ");  
}}  
}
```

Output:

```
Enter the number till which you want the prime number:15  
Prime numbers are:2 3 5 7 11 13
```

Result:

Thus, a java program to generate all the prime numbers between 1 and n, where n is a value supplied by the user is executed successfully and output is verified.

EX 1.D: Java program to find both the largest and smallest number in a list of integers.

Aim:

To write a java program to find both the largest and smallest number in a list of integers.

Algorithm:

1. Input the Array elements.
2. Initialize small= large=arr[0]
3. Repeat From i= 2 to n
4. if(arr[i]>large)
5. large=arr[i]
6. if(arr[i]<small)
7. small=arr[i]
8. Print small and large.

Program:

```
Public class smaLarge {  
    public static void main(String[] args)  
    {  
        int numbers[]=new int[]{15,56,45,6,934,76,98,4321,345,65};  
        int smallest = numbers[0];  
        int biggest=numbers[0];  
        for(int i=1;i<numbers.length;i++)  
        {  
            if(numbers[i]>biggest)  
                biggest=numbers[i];  
            else if (numbers[i] < smallest)  
                smallest= numbers[i];  
        }  
        System.out.println("Largest Number is :"+biggest);  
        System.out.println("SmallestNumber is :"+smallest);  
    }  
}
```

DATE:

NAME:SHWETA DARSHINI P
REGNO:412521104139.

Output:

```
Largest number is :4321  
Smallest number is :6
```

Result:

Thus, a java program to find both the largest and smallest number in a list of integers is executed successfully and output is verified.

**EX 1.E: Java program to find factorial of list of number reading
number input as a command.**

Aim:

To write a java program to find factorial of list of number reading
Input as command.

Algorithm:

- 1.Start
2. Declare Variable n, fact, i
- 3.Read number from User
4. Initialize Variable fact=1 and i=1
- 5.Repeat Until i<=number
 - 5.1fact=fact*i
 - 5.2 i=i+1
- 6.Print fact
- 7.Stop.

Program:

```
class fact{
public static void main(String[] arg){
int[] num=new int[10];
if(arg.length==0)
{
System.out.println("No Command Line argument passed.");
return;
}
for(int i=0;i<arg.length;i++)
num[i]=Integer.parseInt(arg[i]);
for(int i=0;i<arg.length;i++)
{
Int fact=1;
for(int j=1;j<=num[i];j++)
fact*=j;
System.out.println("The Factorial Of"+arg[i]+" is:"+fact);
}
```

DATE:

NAME:SHWETA DARSHINI P
REGNO:412521104139.

```
}}  
}
```

Output:

```
1 6  
The Factorial of 1 is:1  
The Factorial of 6 is:720
```

Result:

Thus, a java program to find factorial of list of number reading input as command is executed successfully and its output is verified.

EX 2: Java program to calculate bonuses for different departments using method overriding.

Aim:

To write a Java program to calculate bonuses for different departments using method overriding.

Algorithm:

1. START
2. Get Basic Pay
3. Bonus for sales=20%*Basic Pay
4. Bonus for marketing=20%* Basic Pay
5. Bonus for HR=50%*Basic Pay
6. Print Basic Pay ,Bonus for sales ,Bonus for marketing ,Bonus for HR
7. STOP

Program:

```
import java.util.*;
abstract class dept
{
    double bp;
    dept(double bpay)
    {
        bp=bpay;
    }
    void disp()
    {
        System.out.println("basic pay="+bp);
    }
    abstract double bonus();
}
class sales extends dept
{
    sales(double bpay)
    {
        super(bpay);
    }
}
```

DATE:

NAME:SHWETA DARSHINI P
REGNO:412521104139.

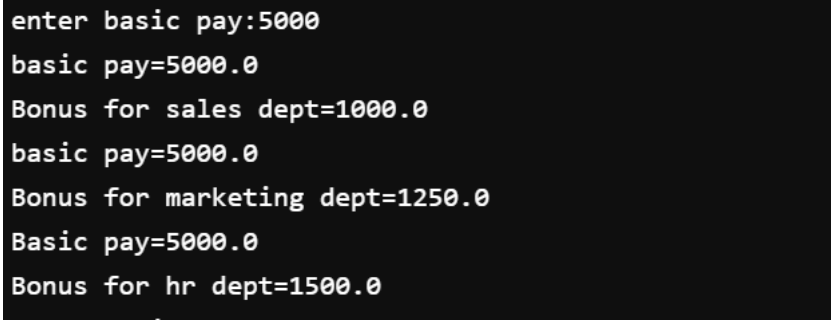
```
}  
public double bonus()  
{  
return(0.20*bp);  
}  
}  
class marketing extends dept  
{  
marketing(double bay)  
{  
super(bpay);  
}  
public double bonus()  
{  
return(0.30*bp);  
}  
}  
class hr extends dept  
{  
hr(double bpay)  
{  
super(bpay);  
}  
public double bonus()  
{  
return(0.50*bp);  
}  
}  
class MethodOverriding  
{  
public static void main(String arg[])  
{  
Scanner sc=new Scanner(System.in);  
System.out.println("enter basic pay");  
double bp=sc.nextdouble();  
sales s=new sales(bp);  
s.disp();
```

DATE:

NAME:SHWETA DARSHINI P
REGNO:412521104139.

```
System.out.println("bonus for sales dept="+s.bonus());  
Marketing m=new marketing(bp);  
m.disp();  
System.out.println("bonus for marketing dept="+m.bonus());  
hr h=new hr(bp);  
h.disp();  
System.out.println("bonus for hr dept="+h.bonus());  
}  
}
```

Output:

A screenshot of a terminal window showing the output of a Java program. The text is as follows:

```
enter basic pay:5000  
basic pay=5000.0  
Bonus for sales dept=1000.0  
basic pay=5000.0  
Bonus for marketing dept=1250.0  
Basic pay=5000.0  
Bonus for hr dept=1500.0
```

Result:

Thus, the Java program to calculate bonus for different departments using method overriding is executed successfully and output is verified.

EX 3: Java program to sort list of elements in ascending and descending order and show the exception handling.

Aim:

To write a Java program to sort list of elements ascending and descending order and show the exception handling.

Algorithm:

1. Start.
2. Get the size of the array and the elements.
3. Using for loops compare the array elements with each other and arrange them in descending order.
4. Print the Array Elements in Descending Order.
5. Using for loops, compare the array elements with each other and arrange the min ascending order.
6. Print the Array Elements in Ascending Order.
7. A try-catch block is used to show exception handling.
8. An exception is thrown when an index value greater than the size of the array is tried to get accessed.
9. Stop.

Program:

```
import java.util.*;
public class Sorting{
public static void main(String[] args)
{
Scanner sc = new Scanner(System.in);
int i,j;
System.out.println("Enter the size of Array");
int n=sc.nextInt();
int[] arr =new int[n];
int temp=0;
System.out.println("Enter the array elements: ");
for(i=0;i<n;i++)
{
arr[i]=sc.nextInt();
```

DATE:

NAME:SHWETA DARSHINI P
REGNO:412521104139.

```
}  
for (i = 0; i < n; i++) {  
    for(j= i+1; j<n; j++)  
    {  
        if(arr[i] < arr[j]) {  
            temp=arr[i];  
            arr[i] = arr[j];  
            arr[j]=temp;  
        }  
    }  
    System.out.println();  
    System.out.println("Descending order: ");  
    for (i = 0; i < n; i++) {  
        System.out.println(arr[i] + " ");  
    }  
    temp=0;  
    for (i = 0; i < n; i++) {  
        for(j= i+ 1;j< n; j++)  
        {if (arr[i]>arr[j]){  
            temp = arr[i];  
            arr[i] = arr[j];  
            arr[j]=temp;  
        }  
    }  
    System.out.println();  
    System.out.println("Ascending order:");  
    for (i = 0; i < n; i++) {  
        System.out.println(arr[i] + " ");  
    }  
    System.out.println();  
    try{  
        System.out.println(arr[n+1]);  
    }  
    catch(ArrayIndexOutOfBoundsException e)  
    {  
        System.out.println(e);  
    }  
}
```

DATE:

NAME:SHWETA DARSHINI P
REGNO:412521104139.

}}

Output:

```
Enter the size of array:5
Enter the array elements:
75
9
1
31
Descending order:
75 31 9 1
Ascending order:
1 9 31 75
```

Result:

Thus, a Java program to sort a list of elements in ascending and descending order and to show the exception handling has been executed successfully.

EX 4: Java program to implement the concept of importing classes from user defined package and creating packages.

Aim:

To write a program to implement the concept of importing classes from user defined packages and creating packages.

Algorithm:

1. Start
2. Create a Package named p1.
3. In p1 package create a class sairam which finds the factorial of a given number.
4. In a new notepad import the package p1 and create a class packdemo in which we create an object for Sairam class and getthefactorialof5.
5. Stop.

Program:

NOTEPAD 1:

```
package p1;  
Public class sairam  
{  
Public int fact(int n)  
{  
if (n<=1)  
return 1;  
else  
return n*fact(n-1);  
}  
}
```

NOTEPAD 2:

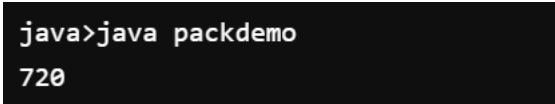
```
import p1.*;  
public class packdemo  
{
```

DATE:

NAME:SHWETA DARSHINI P
REGNO:412521104139.

```
public static void main(String args[])
{
    sairam obj=new sairam();
    System.out.println(obj.fact(6));
}
```

Output:



```
java>java packdemo
720
```

Result:

Thus, a Java program to implement the concept of importing classes from user defined package and creating packages is successfully verified.

EX 5A: Java program that illustrate how the following forms of inheritance supported Single inheritance.

Aim:

To Write a Java Program to illustrate how single inheritance supported.

Algorithm:

1. Start.
2. Create a classA(base class).
3. Declare and initialize variables a and b as 2 and 4 respectively.
4. In classA, create a function fun() which displays the sum of a and b.
5. Create class which inherits class A
6. Inside classB create _function fun1()which Displays the sum of a and b Call Function Fun() in base class
7. Create a A class ex5a which contains the main function.
8. Create an object obj for class B.
9. Callfun1() using obj.
10. Stop.

Program:

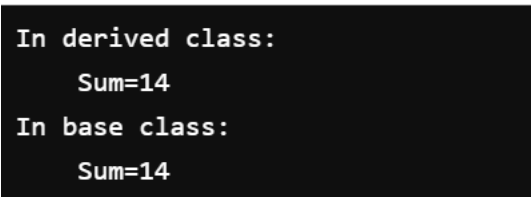
```
class A
{
int a=5,b=9;
void fun()
{
System.out.println("In base class:\n"+"tSum="+a+b);
}
}
class B extends A{
void fun1()
{
System.out.println("In derived class:\n"+"tSum="+a+b);
fun();
}
}
```

DATE:

NAME:SHWETA DARSHINI P
REGNO:412521104139.

```
public class ex5a
{
public static void main(String[] args) {
B obj=new B();
obj.fun1();
}
}
```

Output:



```
In derived class:
    Sum=14
In base class:
    Sum=14
```

Result:

Thus, the Java Program to illustrate how single inheritance is supported is executed successfully and output is verified.

EX 5B: Java program that illustrate how the following forms of inheritance supported Multiple inheritance.

Aim:

To write a Java program to illustrate how multiple inheritance is supported.

Algorithm:

1. Start.
2. Create interface A with function ShowA().
3. Create interface B with function ShowB().
4. Create classC which implements A and B.
5. In classC implement the two functions.
6. Create class ex5b which contains the main function.
7. In the main function create an object obj for class C.
8. Using objc function ShowA() and ShowB().
9. Stop.

Program:

```
interface A
```

```
{
```

```
void ShowA();
```

```
}
```

```
interface B
```

```
{
```

```
void ShowB();
```

```
}
```

```
class C implements A,B
```

```
{
```

```
public void ShowA()
```

```
{
```

```
System.out.println("ShowA() in interfaceA is implemented");
```

```
}
```

```
public void ShowB()
```

```
{
```

```
System.out.println("ShowB()in interfaceB is implemented");
```

DATE:

NAME:SHWETA DARSHINI P
REGNO:412521104139.

```
}  
}  
public class ex5b  
{  
    public static void main(String[] args) {  
        C obj=new C();  
        obj.ShowA();  
        obj.ShowB();  
    }  
}
```

Output:

```
ShowA()in interfaceA is implemented  
ShowB()in interfaceB is implemented
```

Result:

Thus, the Java Program to illustrate how multiple inheritance is supported is executed successfully and output is verified.

EX 5C: Java program that illustrate how the following forms of inheritance supported.

Aim:

To write a Java program to illustrate how multilevel inheritance is supported.

Algorithm:

1. Start
2. Create Class With Function Fun1() that displays "Base class"
3. Create class two which inherits class one and has a function fun2() that displays "Derived class of base class"
4. Create class three which inherits class two and has a function fun3() that displays "Derived Class of derived class"
5. Create A class x5c with main function
6. In the main function create object obj1 for class three
7. Using Obj1 Call the Three Functions
8. Stop.

Program:

```
class one
{
void fun1()
{
System.out.println("Base class");
}
}
class two extends one{
void fun2()
{
System.out.println("Derived class of base class");
}
}
class three extends two
{
void fun3()
```

DATE:

NAME:SHWETA DARSHINI P
REGNO:412521104139.

```
{  
System.out.println("Derived class of derived class");  
}  
}  
public class Main  
{  
public static void main(String[] args) {  
three obj1=new three();  
obj1.fun1();  
obj1.fun2();  
obj1.fun3();  
}  
}
```

Output:

```
Base class  
Derived class of base class  
Derived class of derived class
```

Result:

Thus, the Java program to illustrate how multiple inheritance is supported is executed successfully and output is verified.

EX 5D: Java program that illustrate how the following forms of inheritance supported Hierarchical Inheritance.

Aim:

To write a Java program to illustrate how hierarchical inheritance supported.

Algorithm:

1. Start
2. Create a classA with function display of that prints "Base class"
3. Create a class B that inherits A with function display1() that displays "Derived class1"
4. display1() also displays " Accessing base class function from Derived class 1" and Calls display()
5. Create a classC that inherits A with function display2() that displays"Derived class2"
6. Display2() also displays "Accessing base class function from Derived class 2" and Calls display()
7. Create a class ex5d with main function
8. Inside the main function create objects obj1 and obj2 for classes B and C respectively
9. Call display1() with obj1
10. Call display2() with obj2
11. Stop.

Program:

```
class A
{
void display()
{
System.out.println("Base class\n");
}
}
class B extends A
{
void display1()
```

DATE:

NAME:SHWETA DARSHINI P
REGNO:412521104139.

```
{
System.out.println("Derived class1");
System.out.println("\tAccessing base class function from Derived class 1:");
display();
}
}
class C extends A
{
void display2()
{
System.out.println("Derived class2");
System.out.println("\tAccessing base class function from Derived class 2");
display();
}
}
public class ex5d
{
public static void main(String[] args) {
B obj1=new B();
C obj2=new C();
obj1.display1();
obj2.display2();
}
}
```

Output:

```
Derived class1
    Accessing base class function from Derived class 1
Derived class2
    Accessing base class function from Derived class 2
```

Result:

Thus, the Java program to illustrate how multiple inheritance is supported is executed successfully and output is verified.

EX 6: Java program to demonstrate the use of implementing interfaces

Aim:

To write a program to demonstrate the use of implementing interfaces.

Algorithm:

1. Start
2. Create interface with function ShowA()
3. Create interfaceB with function ShowB()
4. Create classC which implements A and B
5. In classC implement the two functions
6. Create class 5b which contains the main function
7. In the main function create an object obj for classC
8. Using objc function ShowA() and ShowB()
9. Stop

Program:

```
interface AnimalEat{voideat();
}
interface AnimalTravel{void travel();
}
class Animal implements AnimalEat, AnimalTravel{public void eat() {
System.out.println("Animals Eating");
}
public void travel(){
System.out.println("Animal is travelling");
}
}
Public classDemo{
public static void main(String args[]) {
Animal a =newAnimal();
a.eat();
a.travel();
}
}
```

DATE:

NAME:SHWETA DARSHINI P
REGNO:412521104139.

Output:

```
Animals Eating  
Animal is travelling
```

Result:

Thus, the Java program to demonstrate the use of implementing interfaces executed successfully and output is verified.

EX 7: Java program to implement all string operations using interface

Aim:

To Write A java program to implement all string operations using interface.

Algorithm:

1. Start
2. Create an interface strings_interface which declares and initializes strings s, s1, s2, str1, word1, word2 and word4 on which String methods are to be used.
3. Declare void methods findlength(), findcharat(), findsubstring(), concatenate(), findindexof(), checkequality(), checkequalitywithoutcase(), comparingto(), lowercase(), uppercase(), trimword() and replacement()
4. Create class A which implements strings_interface
5. Override and implement all the methods declared in the interface in class A
6. Create class ex7 which extends A
7. Create an object a of A in the main method
8. Using a call all the methods in class A
9. Stop

Program:

```
import java.io.*;
import java.util.*;
interface strings_interface
{
String s="Unbowed Unbent Unbroken";
String s1="one";
String s2="two";
String str1="Blows and Burns";
String word1="YES";
String word2="yes";
String word4="High on honor";
void findlength();
void findcharAt();
void findsubstring();
void concatenate();
void findindexof();
void checkequality();
void checkequalitywithoutcase();
void comparingto();
void lowercase();
void uppercase();
void trimword();
```

DATE:

NAME:SHWETA DARSHINI P

REGNO:412521104139.

```
void replacement();
}
class A implements strings_interface
{
public void findlength()
{
System.out.println("String length of 'Unbowed Unbent Unbroken'="+s.length());
}
public void findcharAt()
{
System.out.println("Character at 3rd position="+s.charAt(3));
}
public void findsubstring()
{
System.out.println("Substring="+s.substring(2,5));
}
public void concatenate()
{
System.out.println("Concatenated String="+s1.concat(s2));
}
public void findindexof()
{
System.out.println("Index of share="+s.indexOf("Unbent"));
}
public void checkequality()
{
Boolean out=s1.equals("one");
System.out.println("Checking Equality of 'one' and 'one':" +out);
}
public void checkequalitywithoutcase()
{
Boolean out="oNe".equalsIgnoreCase("one");
System.out.println("Checking Equality of 'oNe' and 'one' with equals ignore
case:" +out);
}
public void comparingto()
{
int out1=s1.compareTo(s2);
System.out.println("The difference between ASCII values of s1='one' and
s2='two'="+out1);
}
public void lowercase()
{
System.out.println("Changing 'YES' to lowercase:" +word1.toLowerCase());
}
public void uppercase()
```

DATE:

NAME:SHWETA DARSHINI P

REGNO:412521104139.

```
{
System.out.println("Changing 'yes' to uppercase:"+word2.toUpperCase());
}
public void trimword()
{
System.out.println("Trimmed the word"+" "+word4.trim()+"");
}
public void replacement()
{
System.out.println("Orginal String:"+str1);
String str2=str1.replace('s','i');
System.out.println("Replaced 's' with 'i' ->"+str2);
}
}
class ex7 extends A
{
public static void main(String args[])
{
A a= new A();
a.findlength();
a.findcharAt();
a.findsubstring();
a.concatenate();
a.findindexof();
a.checkequality();
a.checkequalitywithoutcase();
a.comparingto();
a.lowercase();
a.uppercase();
a.trimword();
a.replacement();
}
}
```

DATE:

NAME:SHWETA DARSHINI P
REGNO:412521104139.

Output:

```
Y:\Java>java ex7
String length of 'Unbowed Unbent Unbroken'=23
Character at 3rd position=o
Substring=bow
Concatenated String=onetwo
Index of share=8
Checking Equality of 'one' and 'one':true
Checking Equality of 'oNe' and 'one' with equals ignore case:true
The difference between ASCII values of s1='one' and s2='two'=-5
Changing 'YES' to lowercase:yes
Changing 'yes' to uppercase:YES
Trimmed the word'High on honor'
Original String:Blows and Burns
Replaced 's' with 'i' ->Blowi and Burni
```

Result:

Thus the java program to implement all string operations using interface is executed successfully and the output is verified.

DATE:

NAME:SHWETA DARSHINI P
REGNO:412521104139.